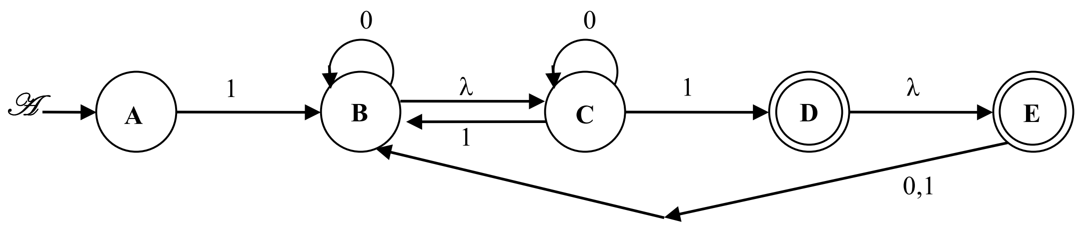
COT 4210 Fall 2016 Sample Problems with Solutions

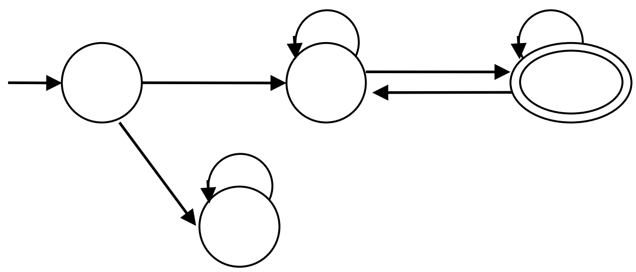
1. Let **L** be defined as the language accepted by the finite state automaton $\mathscr{A}$.



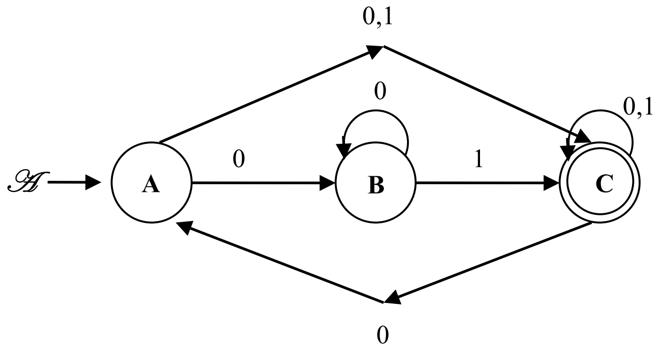
a.) Fill in the following table, showing the λ-closures for each of **A**'s states.

State	**A**	**B**	**C**	**D**	**E**
λ-closure	**{ A }**	**{ B , C }**	**{ C }**	**{ D, E }**	**{ E }**

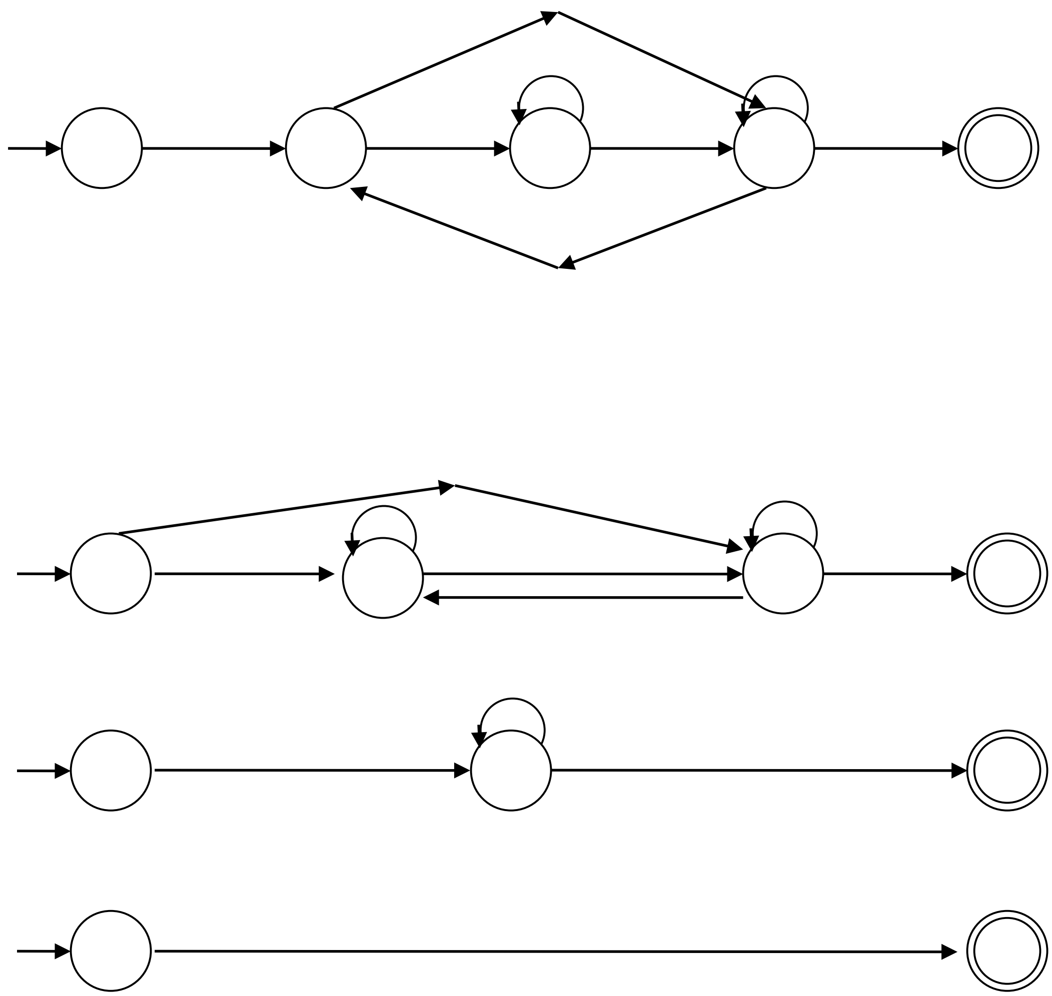
b.) Convert **A** to an equivalent deterministic finite state automaton. Use states like **AC** to denote the subset of states **{A,C}**. Be careful -- λ-closures are important.



2. Let **L** be defined as the language accepted by the finite state automaton $\mathscr{A}$.



Using the technique of ripping (collapsing) states, replacing transition letters by regular expressions, develop the regular expression associated with $\mathscr{A}$ that generates **L**. I have included the diagrams associated with removing states **A**, **B**, then **C**, in that order.



3. Let **L** be recognized by the DFA, $\mathscr{A}$**(Q , Σ , δ , q_0 , F)**, where **|Q|=N**.

Use the Pumping Lemma to show that the following language,
$\mathbf{L = \{ a^n b^m c^t \mid n > m}$ or $\mathbf{n > t}$, and $\mathbf{n, m, t \geq 0}$ **}**, is not regular.

Proof by contradiction:

Assume **L** is regular and let **N** be the number from the P.L. Clearly

$$\mathbf{a^N b^{N-1} c^{N-1} \in L}$$

By P.L., $\mathbf{a^N b^{N-1} c^{N-1} \equiv uvw}$, where $\mathbf{|uv| \leq N}$ and $\mathbf{uw \in L}$, since we can pump as $\mathbf{uv^0w}$. But then, if we compose the expression as the following: $\mathbf{a^{N-|v|}\, a^{|v|}\, b^{N-1}c^{N-1}}$, when we remove **|v| a**'s, via pumping, and we end up with $\mathbf{a^{N-|v|}b^{N-1}c^{N-1}}$ belonging to **L** Since, $\mathbf{|v| > 0}$, the number of **a**'s is less than or equal to the number of **b**'s and **c**'s, which implies $\mathbf{a^{N-|v|}b^{N-1}c^{N-1} \notin L}$, which is a contradiction of our assumption, and therefore **L** is not regular.

4. Analyze the following language, **L**, proving it non-regular by showing that there are an infinite number of equivalence classes formed by the relation $\mathbf{R_L}$ defined by:

$\mathbf{x\ R_L\ y}$ if and only if [$\mathbf{\forall z \in \{a,b,c\}^*}$, $\mathbf{xz \in L}$ exactly when $\mathbf{yz \in L}$].

where $\mathbf{L = \{ a^n b^m c^t \mid n > m > t \}}$.

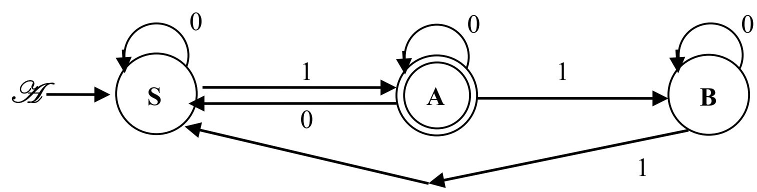
You don't have to present all equivalence classes, but you must demonstrate a pattern that gives rise to an infinite number of classes, along with evidence that these classes are distinct from one another.

Clearly, $\mathbf{a^i b^{i-1} c^{i-2} \in L}$, $\mathbf{a^{i+1} b^{i-1} c^{i-2}}$ and also $\mathbf{a^{i+1} b^i c^{i-1} \in L}$ but $\mathbf{a^i b^i c^{i-1} \notin L}$, which implies, $\mathbf{a^i\ R_L\ a^j}$ iff $\mathbf{i = j}$. Since both $\mathbf{a^i}$ and $\mathbf{a^j}$ are $\mathbf{R_L}$ distinguishable when $\mathbf{i \neq j}$, then there are an infinite number of equivalence classes. Thus **L** is non-regular.

4. Consider the regular grammar **G**:

S	→	**0 S**	**1 A**		
A	→	**0 S**	**0 A**	**1 B**	**λ**
B	→	**1 S**	**0 B**		

a.) Present an automaton $\mathscr{A}$ that accepts the language generated by the **G**:



b) Regular grammars generate the class of regular languages. Regular expressions denote the class of regular sets. The equivalence of these is seen by a proof that every regular set is a regular language and vice versa. The first part of this, that every regular set is a regular language, can be done by first showing that the basis regular sets (**Ø , { λ } , { a | a ∈ Σ }**) are each generated by a regular grammar over the alphabet Σ.

i.) Demonstrate a regular grammar for each of the basis regular sets.

Ø	**G = { {S}, Σ, S, Ø }**
{ λ }	**G = { {S}, Σ, S, {S → λ } }**
{ a }	**G = { {S}. Σ, S, {S → a} }**

Let $\mathbf{L_1}$ be generated by the regular grammar $\mathbf{G_1 = (N_1 , \Sigma , S_1 , P_1)}$ and $\mathbf{L_2}$ be generated by the regular grammar $\mathbf{G_2 = (N_2 , \Sigma , S_2 , P_2)}$, where $\mathbf{N_1 \cap N_2 = Ø}$.

ii.) Present a construction that produces a regular grammar for $\mathbf{L_1 \bullet L_2}$.

$\mathbf{G = \{ N_1 \cup N_2, \Sigma, S_1, P \}}$
$\mathbf{P} = \{ \mathbf{X \rightarrow wS_2} \mid \forall$ rules in $\mathbf{P_1}$ of the form $\mathbf{X \rightarrow w}$, where $\mathbf{X \in N_1}$ and $\mathbf{w \in \Sigma} \} \cup$
$\{ \mathbf{X \rightarrow wY} \mid \forall$ rules in $\mathbf{P_1}$ of the form $\mathbf{X \rightarrow wY}$, where $\mathbf{X, Y \in N_1}$ and $\mathbf{w \in \Sigma} \} \cup \mathbf{P_2}$

Why is the property $\mathbf{N_1 \cap N_2 = Ø}$ needed here?

To prevent rules from the different grammars from mixing with one another when generating the new transition set.

iii.) What remains to be done to show that every regular set is a regular language? Don't do the proof, just state what needs to be done.

Prove closure under union and Kleene*.

5. Present a Mealy Model finite state machine that reads an input **x** ∈ **{0, 1}*** and produces the binary number that represents the result of subtracting **10** from **x** (assumes all numbers are positive, including results). Assume that **x** is read starting with its least significant digit.
Examples: **0010 → 0000; 1000 → 0110; 0001 → 1111** (wrong answer due to going negative)